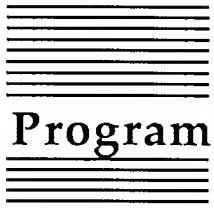
Program Writer

by Alan Bird



Program Writer™

Full Screen BASIC Program Editor
by Alan Bird

Published by
BEAGLE BROS, INC.
6215 Ferris Square, Suite 100
San Diego, CA 92121
619-452-5500 Business Office
619-452-5502 Customer Support
619-558-6151 Pro-Beagle BBS
619-452-6374 Fax

Copyright © 1989 Beagle Bros, Inc.

This manual and the software described in it are copyrighted with all rights reserved. Under the copyright laws, this manual or the software may not be copied, in whole or part, without written consent of Beagle Bros, except in the normal use of the software or to make a backup copy of the software. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased (with all backup copies) may be sold, or given (but not rented or loaned) to another person. Under the law, copying includes translating into another language or format. You may use the software on any computer owned by you, but extra copies cannot be made for this purpose.

LIMITED WARRANTY ON MEDIA AND REPLACEMENT
If you discover physical defects in the manuals distributed with a Beagle Bros product or in the media on which a software product is distributed, Beagle Bros will replace the media or manuals at no charge to you, provided you return the item to be replaced with proof of purchase to Beagle Bros during the 90-day period after you purchased the software.

All implied warranties on the media and manuals, including implied warranties of merchantability and fitness for a particular purpose, are limited in duration to ninety (90) days from the date of the original retail purchase of this product.

Even though Beagle Bros has tested the software and reviewed the documentation, Beagle Bros makes no warranty or representation, either express or implied, with respect to software, its quality, performance, merchantability, or fitness for a particular purpose. As a result, this software is sold "as is," and you the purchaser are assuming the entire risk as to its quality and performance.

In no event will Beagle Bros be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect in the software or its documentation, even if advised of the possibility of such damages. In particular, Beagle Bros shall have no liability for any programs or data stored in or used with Beagle Bros products, including the costs of recovering such programs or data.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Chapter 1 **Welcome to Program Writer 1**

Chapter 2 **Installing the Editor 3**

Chapter 3 **Using the Editor 7**
The Editor 8
Moving the Cursor 10
Editing Commands 13
Adding New Lines 16
Copying and Pasting 17
Finding and Replacing Text 18
Renumbering 20
Miscellaneous Commands 21

Chapter 4 **Macros 25**
Macro Commands 26
Sample Macro File 28

Appendix A **Errors 29**
Invalid number 30
Line overlap 30
Line too big 30
Out of Memory 31

Appendix B **Configuring the Editor 33**

Appendix C **Command Chart 37**

Help! **Customer Support Information 41**

Index 43

Chapter 1

Welcome to Program Writer

PROGRAM WRITER runs on any Apple IIe, IIc, IIc+, or IIgs under both ProDOS and DOS 3.3.

Introduction

PROGRAM WRITER is a utility program that allows you to quickly and effortlessly make changes or add new lines to a BASIC program. Changes are made to the program much like editing a text file with a word processor. PROGRAM WRITER allows you to quickly scroll to the location in your program you want to edit. You can also use a mouse for faster cursor control.

With PROGRAM WRITER you can insert, delete, find and replace, renumber, copy and paste, delete lines, add new lines with automatic line numbering, insert control characters, split a line in two, list all the variables in the program, convert to upper/lower case or copy between programs.

The editor is a memory-resident program, hidden from your own BASIC programs. It is always available for use once it has been installed. Because the program relocates itself, it should be compatible with most other machine language utilities you may be using already.

Chapter 2

Installing the Editor

IMPORTANT: Before using the editor, make a backup copy of the disk. PROGRAM WRITER is not copy-protected. You can use one of the standard copy programs such as COPYA, FILER, or SYSTEM UTILITIES to make the backup.

Installation

There are 3 versions of the editor on the disk. You will normally use the one named EDITOR. A second one named EDITOR.SMALL is for use when little memory is available. Many of the more powerful editing features are not available with EDITOR.SMALL (see the Command Chart for available commands).

The third version, EDITOR.LC, loads into the "language card" or bank-switched memory. Use this version when you have very large programs to edit, or you have several utilities in main memory at the same time. EDITOR.LC requires only a few bytes of main memory.

Under ProDOS, EDITOR.LC requires at least 128K of memory. It loads into the auxiliary bank-switched memory. Under DOS 3.3, it may load into main memory (for a 64K machine) or optionally into auxiliary memory (for a 128K machine). Use the CONFIGURE program to select which bank of memory the DOS 3.3 version will load into (see page 34). The default version loads into main memory. Do not load EDITOR.LC into main memory if you are using a DOS 3.3 mover that relocates DOS up into the "language card".

Each version of the editor runs under DOS 3.3 or ProDOS. Both versions are on the same side of the PROGRAM WRITER disk. When you boot up the disk, you will automatically be in the ProDOS version.

IMPORTANT: Be sure to save your program before installing the editor. If you receive a message telling you to reload your BASIC program, it means that the editor has loaded in over part of your program and destroyed it. This only occurs with very long BASIC programs. The editor initially loads into

memory at 16384 (HEX $4000) and then moves itself as high as it can go in memory.

ProDOS

To use the ProDOS version of the editor, simply boot the disk and select the editor you want from the STARTUP menu. If you have already booted with another ProDOS disk, you can insert the PROGRAM WRITER disk and enter one of the following commands from the "]" prompt:

```
-EDITOR
-EDITOR.SMALL
-EDITOR.LC
```

DOS 3.3

To use the DOS 3.3 version of the editor, boot the disk and select SWITCH TO DOS 3.3 from the STARTUP menu. When the DOS 3.3 menu appears, select an editor. If you have already booted with another DOS 3.3 disk, you can insert the PROGRAM WRITER disk and enter one of the following commands from the "]" prompt:

```
BRUN EDITOR
BRUN EDITOR.SMALL
BRUN EDITOR.LC
```

How to Boot a Disk

Insert the disk to be booted into drive 1 and turn the power on. Another way is to hold down the OPEN-APPLE key and the CONTROL key together, then press and release the RESET key.

Chapter 3

Using the Editor

IMPORTANT: Make it a habit to regularly save your programs on disk. It's good to have a backup version of the program you can return to.

The Editor

To enter the editor, type 2 ampersands (&&) followed by RETURN. The editor screen will contain program statements from the beginning of the BASIC program that is currently in memory. If there is no program in memory, the editing area will be blank.

The Cursor

Most program editing can be done by placing the cursor on the screen where you want to make the change (see MOVING THE CURSOR, page 10), and typing in the corrections to the line.

There are two entry modes: replace and insert. In replace mode, the cursor is a flashing box. Anything you type will replace the character the cursor is on. In insert mode, the cursor is a flashing underscore (_). Characters from the keyboard are inserted in front of the character currently under the cursor. Use the A-E command to switch modes. See page 13 for more information.

Until the BASIC line is actually entered into the program, it can be restored to its original state by pressing ESC. Moving the cursor off of the BASIC line causes that BASIC line to be entered. When this happens, the line cannot be restored using the ESC key.

Screen Width

The editor works in 40 or 80 column text. The default is the active screen mode when the editor is entered. You can switch between 40 and 80 columns with the A-X command (see MISCELLANEOUS COMMANDS, page 22).

Lines

The word "line" has two meanings when using the editor. BASIC lines are lines in a program that start with a line number. Screen lines are any of the 24 lines of characters that you see on your video screen.

Prompt Line

The bottom line of the screen is the prompt line. The number of free bytes you have remaining in memory for your program is displayed there. The dashed line above the prompt line is for separating the prompt line from the main editing area.

Some commands will prompt you for information the editor needs, such as what to look for when using the FIND command. This information is entered in the prompt line.

Many of the editing functions such as insert, delete, and clear to end of line can be used to edit information on the prompt line. Note that unlike Applesoft, everything after the cursor is accepted as part of the input. If, for example, you type in PRINTING as something for the editor to find, and you want to change it to PRINT, you must delete the ING. You cannot simply backspace over the last 3 characters.

Some commands require only a yes or no answer. Press Y or RETURN for yes. To indicate no, press N, SPACE, or ESC.

Control Characters

Embedded control characters in the program are listed in inverse on the screen. Control characters in find-strings or macros also appear in inverse.

Long Lines

A program may contain lines that are too long to edit. Normally, lines this long can only be created by using a utility program that compacts a BASIC program to its smallest size by concatenating lines. A line that is too long to edit from the keyboard is shown by

the editor as a single asterisk (*) following the line number. Most programmers will never see this.

Some long lines are listed by the editor without any spaces between keywords. This keeps the lines short enough that they can still be entered into the program (also see the "No Spaces" command on page 23).

The following commands are all available in the PROGRAM WRITER editor. AppleWorks owners will be pleased that many of the commands are the same as those used by the AppleWorks word processor. If you don't own AppleWorks, don't worry. The commands are very easy to learn.

Definitions

ctrl- To enter a control character, hold down the CONTROL key then press the key following "ctrl-".

A- To enter an OPEN-APPLE command, hold down the OPEN-APPLE key, and press the key following "A-".

Moving the Cursor

Left Arrow

Moves the cursor to the left one space. If the cursor is at the left edge of the screen, it will wrap around to the far right edge.

Right Arrow

Moves the cursor to the right one space. If the cursor is at the right edge of the screen, it will wrap around to the far left edge.

Up Arrow

Moves the cursor up one line. When the cursor is at the top of the screen, the editor will scroll a line at a time until the beginning of the program is reached.

Down Arrow

Moves the cursor down one line. If the cursor is at the bottom of the screen, the editor will scroll to the next line in the BASIC program until the end is reached.

Mouse

You can use the mouse to quickly position the cursor on the screen. Move the cursor off the top or bottom of the screen to scroll.

When you need to scroll many lines, hold down the mouse button. When the cursor is in the top half of the screen and the button is pushed, the screen will scroll up until the button is released. With the cursor in the bottom half, the screen will scroll down. Use the OPEN-APPLE key and the mouse together to scroll a page at a time.

A-Left Arrow

Moves the cursor backward to the previous word on the screen. A word is one or more characters preceded by a space.

A-Right Arrow

Moves the cursor forward to the next word.

A-Up Arrow

Moves the cursor to the top of the screen. If the cursor is already at the top, the screen will scroll down one page.

A-Down Arrow

Moves the cursor to the bottom of the screen. If the cursor is already at the bottom, the screen will scroll up one page.

Return

Hitting the RETURN key anywhere on a line will enter that line and move the cursor to the beginning of the next BASIC line.

RETURN is also used to indicate that you are finished entering something on the prompt line.

Tab

Advances the cursor to the next tab stop. Tab stops are not adjustable. They are preset to every 8th position on the screen.

A-Tab

Moves the cursor left to the previous tab stop.

A-, (A-Comma)

Moves the cursor to the beginning of the BASIC line the cursor is on.

A-. (A-Period)

Moves the cursor to the end of the BASIC line the cursor is on. This is the fastest way to get to the end of the line to add more statements.

You can easily remember these cursor movement commands by the "<" and ">" on the period and comma keys.

A-1 through A-9

Moves the cursor to relative positions within the program. A-1 and A-9 moves the cursor to the beginning or end of the program. A-4 moves it to about the middle. Note that the exact line numbers the cursor moves to will change as the program you are editing shrinks or grows.

A-J

Jumps the cursor to the line number you specify. That BASIC line will appear at the top of the screen. If the line number you entered is not in the program, the first line before the non-existent line number will be displayed.

The editor remembers the last line you jumped to. It becomes the default line number. Next time you use A-J, you won't have to re-enter the line number to jump to the same place.

A-0 (A-Zero)

This command places the cursor on the last line you are editing before you quit the editor to return to BASIC. This is useful if you have done some work on a certain BASIC line, quit the editor and ran the program, and then discovered that the line you had been working on still had some problems. After entering "&&" to return to the editor, press open-apple-0 (zero) and the cursor will jump to that line.

Editing Commands

ESC

Restores the BASIC line you are currently editing to its original form. This is useful if you make errors and would like to start the line all over.

Once the cursor leaves the line, it is entered into the program and you can no longer restore it with the ESC key.

ESC also breaks you out of many commands that require input from the keyboard. For example, if you are entering a string to FIND and you then change your mind, enter ESC to exit the FIND command.

A-E

Switches between insert mode and replace mode. A flashing box cursor ( ) indicates replace mode and a flashing underscore (_) shows insert mode.

Delete

The DELETE key does a backward delete. That is, the DELETE key deletes the character to the left of the cursor.

A-Delete

Deletes the character under the cursor. This is called a forward delete.

A-Y

Deletes everything from the cursor to the end of the BASIC line.

A-O

Overrides any function the following key you enter may have, and inserts it directly into the text. This is necessary for inserting certain control characters such as CONTROL-M (RETURN), CONTROL-H (BACKSPACE) into your program. After entering AO, the cursor will change to an inverse question mark. This lets you know that the following character will be inserted into the line as entered.

ESC will not break out of override mode as it will other commands since you may want to insert an ESC character into the program text.

Here is a useful list of control characters you might want to insert in REM statements or between quotes for PRINT statements. Control characters are shown here preceded by a caret (^).

ctrl-D:	Use this between quotes instead of CHR$(4) for DOS commands: PRINT "^DCATALOG"
ctrl-G:	Causes a beep
ctrl-H:	Backspace
ctrl-J:	Line feed or down arrow
ctrl-K:	Reverse line feed or up arrow
ctrl-L:	Form feed or HOME in 80 columns
ctrl-M:	RETURN
ctrl-U:	Forward arrow
ESC:	Used extensively in printer control strings

A-L

Converts the character at the cursor to lower case. The cursor is also advanced one character, so you can repeatedly enter the A-L command to convert several characters at a time.

A-U

Converts the character at the cursor to upper case. The cursor is also advanced one character, so you can repeatedly enter the A-U command to convert several characters at once.

A-D

Deletes one or more BASIC lines from the program. Place the cursor anywhere on the first line you want to delete, then enter A-D. Move the cursor to the other end of the range of lines you want to delete. To delete one line, leave the cursor where it is. If you are deleting a large number of lines, use the OPEN-APPLE number keys or OPEN-APPLE arrows to quickly position the cursor. Enter RETURN and the editor will display the starting and ending line numbers and ask you if you really want to delete those lines. Enter Y to delete the line, or N to exit without deleting.

> *IMPORTANT: Always think twice before deleting lines. It may be wise to save the program first in case you later decide you need those lines after all.*

A-T

Splits a BASIC line into two lines. Position the cursor where the split is to be made. Everything to the left of the cursor will remain in the same line. Everything to the right will be moved to a new line. The character under the cursor will be deleted, so it is best to split the line with the cursor on the colon(:) that separates two statements.

After entering A-T, you will be prompted for the line number of the new line. The default number is the one immediately following the line that is being split. You can change this to another line number if you want to. Press RETURN and the new line will be entered.

Adding New Lines

A-I

Allows you to insert a new line into the program at any location. When you enter A-I, a blank line is created before the BASIC line the cursor is on. Don't forget to type in a line number before entering the new line.

If you move the cursor to a blank screen line (at the end of the program, for example), it is not necessary to enter A-I. Just type in a line number and enter the BASIC line.

A-A

Allows you to easily add new lines to the program by providing automatic line numbering. After entering A-A, you will be prompted to enter the starting line number for auto numbering. Just enter RETURN if you want the default number provided for you. Then enter the increment between line numbers, or use the default value.

If, for example, you would like the following line numbers to be provided for you: 200, 210, 220...., use 200 as the starting value with an increment of 10.

Once you have entered the starting and incremental values, the first line number will appear at the bottom of the screen. It may be useful before entering A-A to scroll to the point in the program where the text will be added. That way, you will be able to see the program statements that lead up to what you are doing.

After typing in the line, press RETURN and the editor will provide a new line number for the next line. When you are finished with auto numbering, press ESC. You can also exit auto numbering mode by moving the cursor off the line you are adding.

Copying and Pasting

A-C

This command is used to select a range of text to be copied or pasted into other areas of the same program. After entering A-C, use the arrow keys to highlight the text you want to copy, and press RETURN. Text selected with A-C is limited to one BASIC line.

A-P

After using the COPY command to specify the text to be copied, use the PASTE command to paste the text into the program at the current cursor location. You can paste the same text as many times as you want. PASTE can be used in both insert and replace cursor modes.

A-! (works only with EDITOR.LC)

This command is used to copy one or more entire BASIC lines to another program. After entering A-!, use the arrow keys to highlight the BASIC lines you want to copy, and press RETURN.

A-@ (works only with EDITOR.LC)

After using A-! to specify the lines to copy, save the current program. Then, load in the program you want to copy the lines into and press A-@. Be sure that none of the lines you are copying into the new program have the same line numbers. Existing lines with the same numbers will be lost. You can use the RENUMBER command (A-#) to make any necessary changes before pasting in the new lines.

Finding and Replacing Text

A-F

Enter A-F, and you will be prompted to enter something called a find-string. A find-string contains the text you are searching for. It can be as long as the prompt line. This means that in 80 columns, the find-string can be twice as long as in 40 columns. The last find-string you entered will be the default find-string. You can use it again or enter a new one. Use A-Y to remove the default string from the prompt line before typing in a new one.

Once you enter RETURN, the editor will search for and highlight any text in the program which matches your find-string. The editor starts searching from the beginning of the line that the cursor was on when A-F was pressed. To start searching from the beginning of the program, do an A-1 first.

The editor searches within each line individually. Nothing is found that starts in one BASIC line and continues into another.

If no text is found that matches the find-string, the editor will tell you. When a match is found, the text will be highlighted (printed in inverse) and the editor will ask if you want to search again. Pressing Y (or RETURN) for YES will cause the editor to search for the next occurrence. If you answer N (or SPACE) to quit searching, the cursor will stop at the beginning of the text that was found.

No distinction is made between upper and lower case. For example, searching for "Mark" will find "MARK", "mark", or "MaRk".

A-R

Allows you to search for and replace text in the program.

First, enter a find-string as you do with the FIND command. Next, enter the string you would like to replace the find-string with. This is the replace-string. Finally, the editor will ask if you want to replace all occurrences of the find-string. If you answer Y, the editor will automatically search for and replace each occurrence in

the program that matches the find-string. If you answer N, the editor will stop when it finds an occurrence of the find-string and ask you if you want to replace it. Answer Y to replace or N to continue searching. To exit REPLACE mode, press ESC.

A - W

The find and replace commands normally search for the find-string without reference to the surrounding characters. Using A-W you can toggle between word mode and non-word mode. In word mode, the find-string is found only if it is surrounded by spaces, or non-alpha/numeric characters like "!" or "*". Word mode makes it much easier to find variable names and specific line numbers.

For example, searching for the variable I, (in non-word mode) the editor would find I's in "PRINT", "IF", "14", and "INDEX". In word mode, only free-standing I's would be found. While searching for references to line number 10 in non-word mode, the editor would find "100", "210", and "S10". So, use word mode to find line number references.

To determine if you are in word mode or non-word mode, press A-F or A-R. You will see either:

non-word mode:

```
FIND :
REPLACE  :
```

or,

word mode:

```
FIND (WORD) :
REPLACE (WORD) :
```

Renumbering

A-#

Here are some uses of the RENUMBER command that you may not have thought of.

1. Renumbering by an even number such as 10 gives your final product a nice, organized look.

2. In order to insert new lines between two consecutive line numbers (like 20 and 21) you will need to renumber.

3. Renumbering your program in increments of 1, starting at line 0 will help make your program as small as possible.

4. Renumbering a portion of a program allows you to move a range of lines to another location.

To use renumber, place the cursor on the first line that you want to renumber. When renumbering the entire program, place the cursor on the first line of the program (A-1 will do this for you). Press A-#, move the cursor to the last line you want to renumber, and press RETURN. When renumbering the entire program, put the cursor on the last line (using A-9 is the fast way to do this). After selecting the last line to renumber, you will be prompted to verify that those line numbers are correct. If they are not what you wanted, type N to start again.

Next, you will be asked to specify the starting line number for the range of lines you selected. If you are renumbering the entire program and you want the first line to be 100, then just enter 100. If you want a subroutine at line 300 to move to line 2500, then enter 2500 as the starting line number. The line number you enter becomes the default used by the editor next time you renumber.

Finally, you will be asked to specify the line increment. To renumber the entire program starting with line 100 followed by 110, 120, 130..., use the default increment of 10. Press RETURN and the program will be renumbered.

Moving Lines

You can use the RENUMBER command to move one or more lines to another location in the program. In the following example, line 10 is the first line to renumber, line 20 is the last line to renumber, the starting line number is 50 and the increment is 10:

Before Renumbering:

```
10 TEXT  :  REM  THIS LINE WILL MOVE
20 HOME  :  REM  AND THIS ONE, TOO
30 PRINT "HELLO"
40 END
```

After Renumbering:

```
30 PRINT "HELLO"
40 END
50 TEXT  :  REM  THIS WAS LINE 10 BEFORE
60 HOME  :  REM  AND THIS WAS LINE 20
```

If you try to renumber a portion of a program to new line numbers that already exist, you will receive a "LINE OVERLAP" error message.

Miscellaneous Commands

A-Q

This command will return you to BASIC so you can run, save or load a program.

& &

Use the double ampersand followed by RETURN to re-enter the editor.

A-0

Returns you to the last line you were working on before you quit the editor.

A-Z

Removes the editor from memory. This may be necessary if your program requires the memory used by the editor in order to run. This will also recover all memory used by any other memory-resident programs installed after the editor was installed. You will have to re-install the editor to use it again.

A-V

Lists all of the variables in the program alphabetically. Use the command before creating a new variable to find out which variable names have already been used.

Only the first 2 characters of the variable names are listed. String variables are followed by a $, integer variables are followed by a %, and arrays are followed by parenthesis.

Remember that in Applesoft BASIC, only the first 2 characters of a variable name are significant. Don't create a new variable named DOG if the variable list indicates that DO is already being used.

It is all right to reuse variable names of different types. For example, the variable A$ is not the same as A%, A, A$(X), A(X), or A%(X).

A-H

Prints out the current screen on the printer. This feature is useful for printing variable lists (A-V) and macro definitions (A-M). It's also a handy way of printing only a portion of a program.

Inverse characters are printed out underlined on the printer. The output is to the printer in slot 1. If you need to change slots, see "Configuring the Editor" on page 34.

A-X

Switches between 40 and 80 columns. To work on a IIe, 128K is required.

A-<

The A-< (A-shifted comma) command reduces the indentation of the left margin on long BASIC lines that wrap around. Use this command if you would like to get more text on the screen. The "<" indicates the direction the indent will move.

A->

The A-> (A-shifted period) command increases the indentation of the left margin on long BASIC lines that wrap around. Use this command if you would like to make long lines more noticeable. The ">" indicates the direction the indent will move.

A-N

The A-N command toggles between *spaces* and *no-spaces* mode. In *no-spaces* mode there are no spaces displayed between BASIC keywords, variable names, etc. Spaces between words in quotes and following REM and DATA statements are left intact.

Use *no-spaces* mode to get more text in a BASIC line or to see more lines on the screen at one time.

Chapter 4

Macros

A macro is a sequence of characters that the editor can generate automatically by entering a single macro command.

Macro Commands

A-M

The first step in using macros is to create them. Entering A-M gives you a screen with a column of macro command characters down the left side. The command characters indicate which key to press to activate the macro sequence. Move the cursor to the macro you want to add or modify and type in its definition. For example, if you want macro F to generate `FOR I=1 TO 100`, just type it in following the `F:` Then press RETURN to enter the definition.

The maximum length of a macro is 37 characters.

Macros may be nested. That means that a macro may call another macro. To call a macro from within a macro, enter CONTROL-Z followed by the name of the macro. To create a macro that is longer than 37 characters, enter part of the sequence in one macro, then have that macro call another macro containing the remainder of the characters.

Macros can also activate editor commands. An editor command is indicated by CONTROL-A followed by the command character. For example, a macro that would cause a search for the word PRINT would contain the following characters:

```
^AF^AYPRINT^M
```

Which generates:

```
FIND command (A-F)
Clear to the end of line command (A-Y)
"PRINT"
RETURN (CONTROL-M)
```

Control Characters

You will need to use the override command (A-O) to insert control characters such as RETURN (CONTROL-M) into macro definitions.

When you are finished defining macros, press ESC to return to the normal editing mode. If you have just finished editing a macro, remember to hit RETURN to enter the macro before entering ESC. If you don't press RETURN first, ESC will cause the macro to be restored back to what it was, and you'll lose your new macro.

To call a macro, hold down the SOLID-APPLE key (or the Option key on a IIgs) and then press the key for the name of the macro. For example, to activate macro F, hold down the SOLID-APPLE key and enter F.

You can stop a macro in progress at any time by pressing ESC. However, most macros execute so quickly that you won't have time to stop them. But, if you can see that a macro is not doing what you expected, just remember the ESC key.

A-G

Use this command to "get" or load a macro file from disk. Enter the name of the macro file you want to use and it will be loaded into memory.

A-S

Save the current macro file on disk. You will be asked to enter the name of the macro file. You can have any number of macro files as you would like, so unless you want to replace the old file, give it a new name.

By loading and saving your own macro files, you can greatly increase the number of (available) macros. You can customize your own macro sets without having to re-type them every time. Also, you can use the CONFIGURE program to add a default macro file to the editor (see page 34).

> *NOTE: If you have moved DOS 3.3 onto the language card, you cannot use A-S or A-G. However, you can still load a default*

macro file into the editor (see CONFIGURING THE EDITOR, page 34.

Sample Macro File

There is a sample macro file on the PROGRAM WRITER disk called MACROS. Use the A-G command to load it into the editor. You can customize it for your own use if you like. Here is a description of each macro in the MACROS file. Use it for ideas on creating your own customized macros.

A: Appends the line the cursor is on to the end of the previous line. If you receive a "LINE TOO BIG" error message, use the A-P (Paste) function to restore the original second line.
B: Enters the line and moves the cursor to the beginning of the next BASIC line.
C: Generates CHR$( )
D: Generates DATA
E: Moves to the beginning of the last line in the program
F: Finds the next line with a PRINT statement
G: Generates GET
H: Generates HOME
I: Generates IF = THEN
J: Inserts a RETURN (ctrl-M) into the text
K: Generates POKE
L: Converts everything to lower case from the cursor to the end of the program. Press ESC when finished.
M: Generates MID$(, ,)
N: Renumbers the entire program from line 10 with an increment of 10
O: Generates ONERR GOTO
P: Generates PRINT
Q: Deletes everything from the previous word to the end of the line. Keep pressing macro Q to delete the last word in the line.
R: Puts a REM statement at the end of the line
U: Converts everything to upper case from the cursor to the end of the program. Press ESC when finished. Use this macro to convert all text to upper case for a II+ version of

Appendix A

Errors

These are the error messages you might receive from the editor:

Invalid Numbers

This happens when you enter a BASIC line that does not start with a line number or the line number is greater than 63999. You'll also see this message if you specify an invalid number for auto-numbering or renumbering.

You can also get this error during renumbering (A-#). If the new line numbers are greater than 63999, Program Writer will stop and give this error.

Line Overlap

This can occur during renumbering of only a portion of a program. You will get the LINE OVERLAP error if any of the new numbers overlap part of the program that is not being renumbered. This means duplicate or interleaving lines would be the result. This also occurs when pasting lines into another program where the lines would overlap.

Line Too Big

Lines with many statements can become too long. When you attempt to move the cursor from such a line, the editor will tell you that the line is too long. The cursor will be left on the first character in the line that will not fit so you can tell how much is over the limit.

One way out of this problem is to use A-T to break the line in two. That way you won't lose anything.

Out of Memory

Your program is too large to continue editing. This can also happen when you try to renumber your program and there is not enough free memory to build the line number table. Save your program when you get this message. If you are not using EDITOR.SMALL or EDITOR.LC, switch to one of these other editors to get more room for your program.

Appendix B

Configuring the Editor

The program CONFIGURE is a BASIC program on the Program Writer disk that lets you change the editor's command keys, load a default macro file into the editor, specify the default cursor type, set a printer slot, or specify which bank of memory to load the DOS 3.3 version of EDITOR.LC.

If you are used to a another editor that has different command keys, you can change PROGRAM WRITER'S command keys to more familiar ones. Be careful not to change a command key to a character you would normally want to type into a program. Otherwise, you would have to use the A-O (override) command to enter it. Don't use control characters such as CONTROL-M or CONTROL-J as commands, since CONTROL-M is RETURN and CONTROL-J moves the cursor down.

Run the CONFIGURE program and enter the name of the file you want to configure. Usually this will be EDITOR, EDITOR.SMALL, or EDITOR.LC.

When changing command keys, select the command you want to change by highlighting it with the cursor. There are 2 pages of commands. To go to the other one, move the cursor off the top or bottom edge of the screen to the next page. After selecting the command with RETURN, enter the control character or OPEN-APPLE character you want to invoke the command. Specify an OPEN-APPLE character by holding down the OPEN-APPLE key and pressing the character key, or by entering A followed by the character.

To load a macro file into the editor, just type in the name of the file. This will become the default set.

Change the printer slot the editor uses to print the screen by entering a new printer slot (1-7).

To select the default cursor type or the bank of memory for the DOS 3.3 EDITOR.LC, highlight the correct choice, then press RETURN.

After configuring the editor, select "QUIT" from the main menu. The CONFIGURE program will ask you whether or not you want to save the editor with the changes you made. Specify Y for YES if you want the changes saved. Next, you will be prompted for the

file name to save it to. The default file name is the one that was used to load the editor. Press RETURN to use the default name. Enter a new name if you would like to have multiple versions of the editor.

Appendix C

Command Chart

Editor Commands

*	A-A:	Auto line numbering
*	A-C:	Copy (use with Paste)
*	A-D:	Delete lines
	A-E:	Change cursor mode
*	A-F:	Find text
*	A-G:	Get macros
	A-H:	Print screen
	A-I:	Insert line
	A-J:	Jump to line number
	A-L:	Convert to lower case
*	A-M:	Edit macros
	A-N:	No spaces
	A-O:	Override
*	A-P:	Paste (use with Copy)
	A-Q:	Quit editor
*	A-R:	Replace text
*	A-S:	Save macros
*	A-T:	Split into two lines
	A-U:	Convert to upper case
*	A-V:	List program variables
*	A-W:	Toggle Find Word mode
	A-X:	40/80 columns
	A-Y:	Clear to end of line
	A-Z:	Remove editor
	0-Z:	Resume after quit

A-1 thru A-9: Position in program

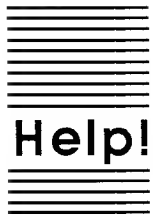
*	A-!:	Copy lines to clipboard
*	A-@:	Copy lines from clipboard
*	A-#:	Renumber
	A-,:	Beginning of line
	A-.:	End of line
	A-<:	Decrease margin
	A->:	Increase margin

ESC:	Restore line, stop macro, quit macro editor, redisplay screen
TAB:	Move to next tab stop
A-TAB:	Move to previous tab stop
RETURN:	Next line, accept input
DELETE:	Delete to left of cursor
A-DELETE:	Delete at cursor
A-Left:	Previous word
A-Right:	Next word
A-Up:	Previous screen
A-Down:	Next screen

Macro Commands

ctrl-A:	Execute editor command
ctrl-Z:	Execute macro

*Not available in EDITOR.SMALL



Customer Support Information

If you have questions or problems that your dealer can't answer, you can contact the Beagle Bros Customer Support Staff for expert assistance.

Before calling, check the instruction manual to see if it contains the information you need. Write down a complete description of the problem, the version number of the software, and the names and version numbers of any other programs you're using in connection with our software.

If you have a modem, you may also receive Technical Support on our 24-hour Customer Support System. The system provides an electronic mail and conferencing system, along with the latest information about product updates and changes.

Customer Support:
(619) 452-5502 8 am to 5 pm, weekdays (Pacific time)

Modem Technical Support:
(619) 558-6151 24 hours, every day

Fax:
(619) 452-6374 24 hours, every day

Or, you can write to:
Beagle Bros, Inc.
6215 Ferris Square, Suite 100
San Diego, CA 92121
Attn: Customer Support

Index

40/80 columns 8, 23
Adding new lines 16
Adjusting margins 23
Asterisk 10
Auto line numbering 16
Backups 4
Booting a disk 5
Clear to end of line 14, 26
Control characters 27
Copying 17
Cursor modes 8
Cursor movement 10
Deleting characters 13
Deleting lines 15
DOS 3.3 5, 27, 34
Escape 8, 13
Finding text 19
Free bytes 9, 31
Insert cursor 8
Inserting lines 17
Installing editor 4
INVALID NUMBER error 30
Language card 4
LINE OVERLAP error 30
LINE TOO BIG error 30
Long lines 9, 30
Lower case conversion 15
Macros 26
Mouse 11
Moving lines 21
OUT OF MEMORY error 31
Override 14, 27
Pasting 17

ProDOS 5
Quitting editor 21
Removing editor 22
Renumbering 20
Replace cursor 8
Replacing text 19
RETURN key 11
Screen width 8, 23
Splitting lines 15
TAB key 12
Upper case conversion 15
Variable list 22

BEAGLE BROS, INC. • 6215 Ferris Square, Suite 100 • San Diego, CA 92121 • (619) 452-5500